UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/735,837	12/15/2003	Michiharu Kudo	JP920020233US1	5521

71034 7590 12/05/2008
LAW OFFICE OF DONALD L. WENSKAY
P.O. Box 7206
Ranco Santa Fe, CA 92067

EXAMINER	
DWIVEDI, MAHESH H	
ART UNIT	PAPER NUMBER
2168	
MAIL DATE	DELIVERY MODE
12/05/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/735,837	**Applicant(s)** KUDO ET AL.	
	Examiner MAHESH H. DWIVEDI	**Art Unit** 2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *06 December 2007*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-21* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *15 December 2003* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)☒ Some * c)□ None of:

1.☒ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *12/06/2007*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

Information Disclosure Statement

1. The information disclosure statements (IDS) submitted on 12/06/2007 has been received, entered into the record, and considered. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner.

Priority

2. Receipt is acknowledged of papers submitted under 35 U.S.C. 119(a)-(d), which papers have been placed of record in the file.

Remarks

3. Receipt of Applicant's Amendment filed on 12/06/2007 is acknowledged. The amendment includes the amending of the specification, the amending of claims 16, 10, 11, 16, 17, 18, 19, and the addition of claim 21.

Specification

4. The objections raised in the office action mailed on 09/07/2006 have been overcome by Applicant's amendment received on 12/06/2007.

Claim Objections

5. The objections raised in the office action mailed on 09/07/2006 have been overcome by Applicant's amendment received on 12/06/2007.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

7. Claim 1 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The examiner specifically points to **"**a **logic operation unit** for deciding access rights in database retrieval using the path expression by performing **logic operations"** as being directed towards nonstatutory subject matter.

The claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, function descriptive material *per se*.

Claims 2-5 are rejected for incorporating the deficiencies of independent claim 1.

Claim 2 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The examiner specifically points to **"**wherein the **logic operation unit** performs decision of the access right**"** as being directed towards nonstatutory subject matter.

The claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, function descriptive material *per se*.

Claim 3 is rejected for incorporating the deficiencies of independent claim 2.

Claim 11 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The examiner specifically points to **"a logic operation unit** for deciding access rights in database retrieval using the path expression by performing **logic operations** related to the query automaton generated by the query automaton generation unit and the access control automaton generated by the access control automaton generation unit**"** as being directed towards nonstatutory subject matter.

The claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, function descriptive material *per se*.

Claim 12 is rejected for incorporating the deficiencies of independent claim 11.

Claim 15 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The examiner specifically points to **"performing logic operations** related to the query automaton and the access control automaton, which are stored in the predetermined storage means, and deciding an access right in database retrieval using the path expression without checking the XML documents stored in the database**"** as being directed towards nonstatutory subject matter.

The claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, function descriptive material *per se*.

Claim 17 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The examiner specifically points to **"logic operation means** for deciding access rights in database retrieval using the path expression by **performing logic operations** related to the generated query automaton and access control automaton**"** as being directed towards nonstatutory subject matter.

The claims lack the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. As such, they fail to fall within a statutory category. They are, at best, function descriptive material *per se*.

Claim 18 is rejected for incorporating the deficiencies of independent claim 17.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

10. Claims 6, 8-10, 13-14, 16, and 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Damiani et al.** (Article entitled "A Fine-Grained Access Control System for XML Documents", dated May 2002) and further in view of **Deo et al.** (U.S. Patent 6,970,891).

11. Regarding claim 6, **Damiani** teaches an information processor comprising:

A) a path table control unit for controlling a path table describing paths of a data file stored in the database (Pages 183 and 186, Figure 5); and

B) an access right decision unit for selecting a predetermined path in the path table controlled by the path table control unit by a path expression describing a retrieval condition for the database (Page 186, Figure 5);

C) applying the access control policy describing the access control rules (Pages 183 and 186, Figure 5); and

D) deciding an access right in database retrieval by the path expression with respect to the predetermined path (Pages 183 and 186, Figure 5).

The examiner notes that **Damiani** teaches **"a path table control unit for controlling a path table describing paths of a data file stored in the database"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"an access right decision unit for selecting a predetermined path in the path table**

controlled by the path table control unit by a path expression describing a retrieval condition for the database" as "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations). The examiner further notes that Figure 5 of Damiani clearly shows different access conditions for different paths in a database for queries from users. The examiner further notes that **Damiani** teaches **"applying the access control policy describing the access control rules"** as "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations) and "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations). The examiner further notes that **Damiani** teaches **"deciding an access right in database retrieval by the path expression with respect to the predetermined path"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations).

Damiani does not explicitly teach:

E) <u>said selecting, applying and deciding being performed prior to retrieving said structured document in said database</u>.

Deo, however, teaches **"<u>said selecting, applying and deciding being performed prior to retrieving said structured document in said database</u>"** as "One or more applications may access individual volatile files 122 using the APIs 202. For instance, suppose an application would like to open a volatile file named "File1". In an implementation using Windows-brand operating system, the application calls a function named "SewCreateFile( )", which opens an existing file or creates a new file if one does not exist. In response to this function call, the file system 118 initially visits the ACL table 204 to determine whether the application has sufficient authorization to access the requested file. This is represented by flow arrow 230 in FIG. 2. Assuming the application is authorized, the file system 118 proceeds to the memory region directory 206 to determine which memory region holds the requested file (flow arrow 232). In this example, the directory 206 indicates that the file "File1" is located in RAM 106. The memory region directory 206 directs reference to the RAM-based file allocation table 212 (flow arrow 234), which provides physical location information to locate the volatile file 122(1) within RAM 106 (flow arrow 236)" (Column 5, lines 25-44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Deo's** would have allowed **Damiani's** to provide a method to protect files against from rouge or malicious agents, as noted by **Deo** (Column 1, lines 51-54).

Regarding claim 7, **Damiani** further teaches an information processor comprising:

A) an access control automaton generation unit for generating an access control automaton from the access control policy in which the access control rule is described (Pages 185-186); and

B) wherein the access right decision unit selects the predetermined path by use of the query automaton generated by the query automaton generation unit; and decides an access right to the predetermined path by use of the access control automaton generated by the access control automaton generation unit (Page 191).

The examiner notes that **Damiani** teaches **"an access control automaton generation unit for generating an access control automaton from the access control policy in which the access control rule is described"** as "An access authorization a ε Auth is a five-tuple of the form: <subject, object, action, sign, type>" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that it is common knowledge that an automaton is a five-tuple with states, symbols, and transition states. The examiner further notes that **Damiani** teaches **"wherein the access right decision unit selects the predetermined path by use of the query automaton generated by the query automaton generation unit; and decides an access right to the predetermined path by use of the access control automaton**

generated by the access control automaton generation unit" as "In particular, the final sign finlabel of each node n is determined as the result of operation ⊕ between the sign field of components of array n.veclabel considered in their priority order: LDH (local hard), RDH (recursive hard), L (local), R (recursive), LD (local, schema level), RD (recursive, schema level), LS (local soft), and RS (recursive soft)" (Page 191, Section 6.1: Document Tree Labeling).

Damiani and **Deo** do not explicitly teach:

C) a query automaton generation unit for generating a query automaton from a path expression in which a retrieval condition for the database is described.

Murata, however, teaches **"a query automaton generation unit for generating a query automaton from a path expression in which a retrieval condition for the database is described"** as "A selection query is select (e1, e2) where e1 is a hedge regular expression and e2 is a pointed hedge representation" (Pages 132-133, Section 6.1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Murata's** would have allowed **Damiani's** and **Deo's** to provide a method to allow for further processing of output relations by enabling schema translations via queries for xml documents, as noted by **Murata** (Pages 126-127, Section 1).

Regarding claim 8, **Damiani** further teaches an information processor comprising:

A) a path expression extraction unit for extracting the path expressions from a query expression specifying a retrieval method for the database (Pages 185-186).

The examiner notes that **Damiani** teaches **"a path expression extraction unit for extracting the path expressions from a query expression specifying a retrieval method for the database"** as "object is either a URI in Obj or is of the form URI:PE, where URI ε Obj and PE is a path expression on the tree of document URI" (Page 185, Section 5.2: Access Authorizations).

Regarding claim 9, **Damiani** further teaches an information processor comprising:

A) a query expression access right decision unit for deciding access rights in the database retrieval by the query expression based on decision results of access rights, which are obtained by the access right decision unit, for the individual path expressions extracted from the query expression (Pages 190-191).

The examiner notes that **Damiani** teaches **"a query expression access right decision unit for deciding access rights in the database retrieval by the query expression based on decision results of access rights, which are obtained by the access right decision unit, for the individual path expressions extracted from the query expression"** as "The value of n.veclabel[t].sign can be "+" for permission, "-" for denials, and "ε" for no authorization" (Page 188, Section 6.1: Document Tree Labeling) and "Signs + and – must then be mapped to the other two values, namely 1 (true) and ½ (indeterminate" (Page 190, Section 6.1: Document Tree Labeling) and "As a result of the labeling process, the value of finlabel for each node n contains the sign, if any, reflecting whether the node can be accessed (+) or not (-)" (Page 191, Section 6.2: Transformation Process).

Regarding claim 10, **Damiani** teaches a database retrieval system comprising:

A) a database storing an XML document (Page 171); and

B) <u>a preliminary</u> access rights analysis device which decides, based on path expressions describing retrieval conditions used in retrieval for the database and an access control policy describing access control rules, to which one of 1) always permitted, 2) always denied, and 3) indeterminate an access right in the database retrieval using the path expressions corresponds (Pages 188, 190).

The examiner notes that **Damiani** teaches **"a database storing an XML document"** as "The rationale for our approach is defining an XML markup for a set of security elements describing the protection requirements of XML documents" (Page 171, Section 1: Introduction). The examiner further notes that **Damiani** teaches **"<u>a</u>**

<u>preliminary</u> access rights analysis device which decides, based on path expressions describing retrieval conditions used in retrieval for the database and an access control policy describing access control rules, to which one of 1) always permitted, 2) always denied, and 3) indeterminate an access right in the database retrieval using the path expressions corresponds" as "The value of n.veclabel[t].sign can be "+" for permission, "-" for denials, and "ε" for no authorization" (Page 188, Section 6.1: Document Tree Labeling) and "Signs + and – must then be mapped to the other two values, namely 1 (true) and ½ (indeterminate" (Page 190, Section 6.1: Document Tree Labeling).

Damiani does not explicitly teach:

C) <u>said preliminary access rights analysis device deciding said access rights without retrieving said XML document</u>.

Deo, however, teaches **"<u>said preliminary access rights analysis device deciding said access rights without retrieving said XML document</u>"** as "One or more applications may access individual volatile files 122 using the APIs 202. For instance, suppose an application would like to open a volatile file named "File1". In an implementation using Windows-brand operating system, the application calls a function named "SewCreateFile( )", which opens an existing file or creates a new file if one does not exist. In response to this function call, the file system 118 initially visits the ACL table 204 to determine whether the application has sufficient authorization to access the requested file. This is represented by flow arrow 230 in FIG. 2. Assuming the application is authorized, the file system 118 proceeds to the memory region directory 206 to determine which memory region holds the requested file (flow arrow 232). In this example, the directory 206 indicates that the file "File1" is located in RAM 106. The memory region directory 206 directs reference to the RAM-based file allocation table 212 (flow arrow 234), which provides physical location information to locate the volatile file 122(1) within RAM 106 (flow arrow 236)" (Column 5, lines 25-44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching

Deo's would have allowed **Damiani's** to provide a method to protect files against from rouge or malicious agents, as noted by **Deo** (Column 1, lines 51-54).

Regarding claim 13, **Damiani** further teaches a database retrieval system comprising:

A) a path table control unit for controlling a path table describing paths of a data file stored in the database (Pages 183 and 186, Figure 5); and

B) an access right decision unit for selecting a predetermined path in the path table controlled by the path table control unit by a path expression describing a retrieval condition for the database (Page 186, Figure 5);

C) applying the access control policy describing the access control rules (Pages 183 and 186, Figure 5); and

D) deciding an access right in database retrieval by the path expression with respect to the predetermined path (Pages 183 and 186, Figure 5).

The examiner notes that **Damiani** teaches **"a path table control unit for controlling a path table describing paths of a data file stored in the database"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"an access right decision unit for selecting a predetermined path in the path table controlled by the path table control unit by a path expression describing a retrieval condition for the database"** as "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations). The examiner further notes that Figure 5 of Damiani clearly shows different access conditions for different paths in a database for queries from users. The examiner further notes that **Damiani** teaches **"applying the access control policy describing the access control rules"** as "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations) and "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations). The examiner further notes that **Damiani** teaches **"deciding an access right in database retrieval by the path expression with respect to the predetermined path"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations).

Regarding claim 14, **Damiani** further teaches a database retrieval system comprising:

A) a path expression extraction unit for extracting the path expressions from a query expression specifying a retrieval method for the database (Pages 185-186); and

B) a query expression access right decision unit for deciding access rights in the database retrieval by the query expression based on decision results of access rights, which are obtained by the access right decision unit, for the individual path expressions extracted from the query expression (Pages 190-191).

The examiner notes that **Damiani** teaches **"a path expression extraction unit for extracting the path expressions from a query expression specifying a retrieval method for the database"** as "object is either a URI in Obj or is of the form URI:PE, where URI ε Obj and PE is a path expression on the tree of document URI" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"a query expression access right decision unit for deciding access rights in the database retrieval by the query expression based on decision results of access rights, which are obtained by the access right decision unit, for the individual path expressions extracted from the query expression"** as "The value of n.veclabel[t].sign can be "+" for permission, "-" for denials, and "ε" for no authorization"

(Page 188, Section 6.1: Document Tree Labeling) and “Signs + and – must then be mapped to the other two values, namely 1 (true) and ½ (indeterminate” (Page 190, Section 6.1: Document Tree Labeling) and “As a result of the labeling process, the value of finlabel for each node n contains the sign, if any, reflecting whether the node can be accessed (+) or not (-)” (Page 191, Section 6.2: Transformation Process).

Regarding claim 16, **Damiani** teaches an access rights analysis method comprising:

A) selecting a predetermined path from a path table (Page 186, Figure 5);

B) which is stored in a predetermined storage means and describes paths of a data file stored in the database (Pages 183 and 186, Figure 5);

C) by a path expression describing a retrieval condition for the database (Pages 181-182, 185-186); and

D) applying an access control policy describing access control rules (Pages 183 and 186, Figure 5);

E) deciding an access right in database retrieval by the path expression with respect to the predetermined path (Pages 183 and 186, Figure 5)

The examiner further notes that **Damiani** teaches **“selecting a predetermined path from a path table”** as “Figure 5 lists the resulting authorizations” (Page 186, Section 5.2: Access Authorizations). The examiner further notes that Figure 5 of Damiani clearly shows different access conditions for different paths in a database for queries from users. The examiner further notes that **Damiani** teaches **“which is stored in a predetermined storage means and describes paths of a data file stored in the database”** as “Authorizations specified on an element can be defined as applicable to the element’s attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)” (Page 183, Section 5.1: Basic Features of the Access Authorizations) and “Figure 5 lists the resulting authorizations” (Page 186, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **“by a path expression describing a retrieval condition for the database”** as “A path expression l1/l2/…/ln on a document tree represents all the attributes named ln that can be reached by descending the document tree along the sequence of nodes named l1,l2,…,ln-1" (Page 181, Section 4: Authorization Objects" and "object is either a URI in Obj or is of the form URI:PE, where URI ε Obj and PE is a path expression on the tree of document URI" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"applying an access control policy describing access control rules"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"deciding an access right in database retrieval by the path expression with respect to the predetermined path"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations).

Damiani does not explicitly teach:

E) without checking the data file stored in the database.

Deo, however, teaches **"without checking the data file stored in the database"** as "One or more applications may access individual volatile files 122 using the APIs 202. For instance, suppose an application would like to open a volatile file named "File1". In an implementation using Windows-brand operating system, the application calls a function named "SewCreateFile( )", which opens an existing file or creates a new file if one does not exist. In response to this function call, the file system 118 initially visits the ACL table 204 to determine whether the application has sufficient authorization to access the requested file. This is represented by flow arrow 230 in FIG. 2. Assuming the application is authorized, the file system 118 proceeds to the memory region directory 206 to determine which memory region holds the requested file (flow arrow 232). In this example, the directory 206 indicates that the file "File1" is located in

RAM 106. The memory region directory 206 directs reference to the RAM-based file allocation table 212 (flow arrow 234), which provides physical location information to locate the volatile file 122(1) within RAM 106 (flow arrow 236)" (Column 5, lines 25-44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Deo's** would have allowed **Damiani's** to provide a method to protect files against from rouge or malicious agents, as noted by **Deo** (Column 1, lines 51-54).

Regarding claim 19, **Damiani** further teaches a program comprising:

A) a path table control means for controlling a path table describing paths of a data file stored in the database (Pages 183 and 186, Figure 5); and

B)) an access right decision means for selecting a predetermined path in the path table controlled by the path table control unit by a path expression describing a retrieval condition for the database (Page 186, Figure 5);

C) applying an access control policy describing access control rules (Pages 183 and 186, Figure 5); and

D) deciding the presence of an access right in database retrieval by the path expression with respect to the predetermined path (Pages 183 and 186, Figure 5).

The examiner notes that **Damiani** teaches **"a path table control means for controlling a path table describing paths of a data file stored in the database"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"an access right decision means for selecting a predetermined path in the path table controlled by the path table control unit by a path expression describing a retrieval condition for the database"** as "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations). The examiner further notes that Figure 5 of Damiani clearly shows different access conditions for different paths in a database for queries from users. The examiner further notes that **Damiani** teaches **"applying an access control policy describing access control rules"** as "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations) and "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations). The examiner further notes that **Damiani** teaches **"deciding the presence of an access right in database retrieval by the path expression with respect to the predetermined path"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations).

Damiani does not explicitly teach:

E) without accessing said data file.

Deo, however, teaches **"without accessing said data file"** as "One or more applications may access individual volatile files 122 using the APIs 202. For instance, suppose an application would like to open a volatile file named "File1". In an implementation using Windows-brand operating system, the application calls a function named "SewCreateFile( )", which opens an existing file or creates a new file if one does not exist. In response to this function call, the file system 118 initially visits the ACL table 204 to determine whether the application has sufficient authorization to access the requested file. This is represented by flow arrow 230 in FIG. 2. Assuming the application is authorized, the file system 118 proceeds to the memory region directory 206 to determine which memory region holds the requested file (flow arrow 232). In this example, the directory 206 indicates that the file "File1" is located in RAM 106. The memory region directory 206 directs reference to the RAM-based file allocation table 212 (flow arrow 234), which provides physical location information to locate the volatile file 122(1) within RAM 106 (flow arrow 236)" (Column 5, lines 25-44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Deo's** would have allowed **Damiani's** to provide a method to protect files against from rouge or malicious agents, as noted by **Deo** (Column 1, lines 51-54).

Regarding claim 20, **Damiani** further teaches a program comprising:
A) a path expression extraction means for extracting the path expressions from a query expression specifying a retrieval method for the database (Pages 185-186); and
B) a query expression access right decision means for deciding access rights in the database retrieval by the query expression based on decision results of access rights for the individual path expressions extracted from the query expression (Pages 190-191).

The examiner notes that **Damiani** teaches **"a path expression extraction means for extracting the path expressions from a query expression specifying a retrieval method for the database"** as "object is either a URI in Obj or is of the form URI:PE, where URI ε Obj and PE is a path expression on the tree of document URI" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"a query expression access right decision means for deciding access rights in the database retrieval by the query expression based on decision results of access rights for the individual path expressions extracted from the query expression"** as "The value of n.veclabel[t].sign can be "+" for permission, "-" for denials, and "ε" for no authorization" (Page 188, Section 6.1: Document Tree Labeling) and "Signs + and – must then be mapped to the other two values, namely 1 (true) and ½ (indeterminate" (Page 190, Section 6.1: Document Tree Labeling) and "As a result of the labeling process, the value of finlabel for each node n contains the sign, if any, reflecting whether the node can be accessed (+) or not (-)" (Page 191, Section 6.2: Transformation Process).

12. Claims 1-5, 7, 11-12, 15, and 17-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Damiani et al.** (Article entitled "A Fine-Grained Access Control

System for XML Documents", dated May 2002) in view of **Deo et al.** (U.S. Patent 6,970,891) as applied to claims 6, 8-10, 13-14, 16, and 19-20, and further in view of **Murata** (Article entitled "Extended Path Expressions for XML", dated 04/29/2001).

13. Regarding claim 1, **Damiani** teaches an information processor comprising:

B) an access control automaton generation unit for generating an access control automaton from an access control policy in which an access control rule is described (Pages 185-186); and

C) a logic operation unit for deciding access rights in database retrieval using the path expression by performing logic operations related to the query automaton generated by the query automaton generation unit and the access control automaton generated by the access control automaton generation unit (Page 191).

The examiner notes that **Damiani** teaches **"an access control automaton generation unit for generating an access control automaton from an access control policy in which an access control rule is described"** as "An access authorization a ε Auth is a five-tuple of the form: <subject, object, action, sign, type>" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that it is common knowledge that an automaton is a five-tuple with states, symbols, and transition states. The examiner further notes that **Damiani** teaches **"a logic operation unit for deciding access rights in database retrieval using the path expression by performing logic operations related to the query automaton generated by the query automaton generation unit and the access control automaton generated by the access control automaton generation unit"** as "In particular, the final sign finlabel of each node n is determined as the result of operation ⊕ between the sign field of components of array n.veclabel considered in their priority order: LDH (local hard), RDH (recursive hard), L (local), R (recursive), LD (local, schema level), RD (recursive, schema level), LS (local soft), and RS (recursive soft)" (Page 191, Section 6.1: Document Tree Labeling).

Damiani does not explicitly teach:

D) without accessing said data file stored in said database.

Deo, however, teaches **"without accessing said data file stored in said database"** as "One or more applications may access individual volatile files 122 using the APIs 202. For instance, suppose an application would like to open a volatile file named "File1". In an implementation using Windows-brand operating system, the application calls a function named "SewCreateFile( )", which opens an existing file or creates a new file if one does not exist. In response to this function call, the file system 118 initially visits the ACL table 204 to determine whether the application has sufficient authorization to access the requested file. This is represented by flow arrow 230 in FIG. 2. Assuming the application is authorized, the file system 118 proceeds to the memory region directory 206 to determine which memory region holds the requested file (flow arrow 232). In this example, the directory 206 indicates that the file "File1" is located in RAM 106. The memory region directory 206 directs reference to the RAM-based file allocation table 212 (flow arrow 234), which provides physical location information to locate the volatile file 122(1) within RAM 106 (flow arrow 236)" (Column 5, lines 25-44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Deo's** would have allowed **Damiani's** to provide a method to protect files against from rouge or malicious agents, as noted by **Deo** (Column 1, lines 51-54).

Damiani and **Deo** do not explicitly teach:

A) a query automaton generation unit for generating a query automaton from a path expression in which a retrieval condition for the database is described.

Murata, however, teaches **"a query automaton generation unit for generating a query automaton from a path expression in which a retrieval condition for the database is described"** as "A selection query is select (e1, e2) where e1 is a hedge regular expression and e2 is a pointed hedge representation" (Pages 132-133, Section 6.1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Murata's** would have allowed **Damiani's** and **Deo's** to provide a method to allow for further processing of output relations by enabling schema translations via queries for xml documents, as noted by **Murata** (Pages 126-127, Section 1).

Regarding claim 2, **Damiani** teaches an information processor comprising:

A) wherein the logic operation unit performs decision of the access right (Page 191).

The examiner notes that **Damiani** teaches **"wherein the logic operation unit performs decision of the access right"** as "In particular, the final sign finlabel of each node n is determined as the result of operation ⊕ between the sign field of components of array n.veclabel considered in their priority order: LDH (local hard), RDH (recursive hard), L (local), R (recursive), LD (local, schema level), RD (recursive, schema level), LS (local soft), and RS (recursive soft)" (Page 191, Section 6.1: Document Tree Labeling).

Damiani and **Deo** do not explicitly teach:

B) a schema automaton generation unit for generating a schema automaton from a schema showing a structure of the data file stored in the database; and

C) in consideration for the schema automaton generated by the schema automaton generation unit.

Murata, however, teaches **"a schema automaton generation unit for generating a schema automaton from a schema showing a structure of the data file stored in the database"** and **"in consideration for the schema automaton generated by the schema automaton generation unit"** as "Schema transformation is effected by first creating intersection hedge automata which stimulate the match identifying hedge automata and the input schemata, and then transforming the intersection hedge automata as appropriate to the query operation" (Pages 127, Section 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Murata's** would have allowed **Damiani's** and **Deo's** to provide a method to allow for further processing of output relations by enabling schema translations via queries for xml documents, as noted by **Murata** (Pages 126-127, Section 1).

Regarding claim 3, **Damiani** teaches an information processor comprising:

A) a path table control unit for controlling path table describing paths of the data file stored in the database (Pages 183 and 186).

The examiner notes that **Damiani** teaches **"a path table control unit for controlling path table describing paths of the data file stored in the database"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations).

Damiani and **Deo** do not explicitly teach:

B) wherein the schema automaton generation unit generates the schema automaton from the path table controlled by the path table control unit.

Murata, however, teaches **"wherein the schema automaton generation unit generates the schema automaton from the path table controlled by the path table control unit"** as "Schema transformation is effected by first creating intersection hedge automata which stimulate the match identifying hedge automata and the input schemata, and then transforming the intersection hedge automata as appropriate to the query operation" (Pages 127, Section 1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Murata's** would have allowed **Damiani's** and **Deo's** to provide a method to allow for further processing of output relations by enabling schema translations via queries for xml documents, as noted by **Murata** (Pages 126-127, Section 1).

Regarding claim 4, **Damiani** further teaches an information processor comprising:

A) a path expression extraction unit for extracting the path expressions from a query expression specifying a retrieval method for the database (Pages 181-182, 185-186).

The examiner notes that **Damiani** teaches **“a path expression extraction unit for extracting the path expressions from a query expression specifying a retrieval method for the database”** as “A path expression l1/l2/…/ln on a document tree represents all the attributes named ln that can be reached by descending the document tree along the sequence of nodes named l1,l2,…,ln-1” (Page 181, Section 4: Authorization Objects” and “object is either a URI in Obj or is of the form URI:PE, where URI ε Obj and PE is a path expression on the tree of document URI” (Page 185, Section 5.2: Access Authorizations).

Regarding claim 5, **Damiani** further teaches an information processor comprising:

A) a query expression access right decision unit for deciding access rights in the database retrieval by the query expression based on decision results of access rights, which are obtained by the access right decision unit, for the individual path expressions extracted from the query expression (Pages 190-191).

The examiner further notes that **Damiani** teaches **“a query expression access right decision unit for deciding access rights in the database retrieval by the query expression based on decision results of access rights, which are obtained by the access right decision unit, for the individual path expressions extracted from the query expression”** as “The value of n.veclabel[t].sign can be “+” for permission, “-“ for denials, and “ε” for no authorization” (Page 188, Section 6.1: Document Tree Labeling) and “Signs + and – must then be mapped to the other two values, namely 1 (true) and ½ (indeterminate” (Page 190, Section 6.1: Document Tree Labeling) and “As a result of the labeling process, the value of finlabel for each node n contains the sign, if any, reflecting whether the node can be accessed (+) or not (-)” (Page 191, Section 6.2: Transformation Process).

Regarding claim 11, **Damiani** further teaches a database retrieval system comprising:

A) an access control automaton generation unit for generating an access control automaton from the access control policy in which an access control rule is described (Pages 185-186); and
B) a logic operation unit for deciding access rights in database retrieval using the path expression by performing logic operations related to the query automaton generated by the query automaton generation unit and the access control automaton generated by the access control automaton generation unit (Page 191).

The examiner notes that **Damiani** teaches **"an access control automaton generation unit for generating an access control automaton from the access control policy in which an access control rule is described"** as "An access authorization a ε Auth is a five-tuple of the form: <subject, object, action, sign, type>" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that it is common knowledge that an automaton is a five-tuple with states, symbols, and transition states. The examiner further notes that **Damiani** teaches **"a logic operation unit for deciding access rights in database retrieval using the path expression by performing logic operations related to the query automaton generated by the query automaton generation unit and the access control automaton generated by the access control automaton generation unit"** as "In particular, the final sign finlabel of each node n is determined as the result of operation ⊕ between the sign field of components of array n.veclabel considered in their priority order: LDH (local hard), RDH (recursive hard), L (local), R (recursive), LD (local, schema level), RD (recursive, schema level), LS (local soft), and RS (recursive soft)" (Page 191, Section 6.1: Document Tree Labeling).

Damiani and **Deo** do not explicitly teach:

C) wherein the access rights analysis device includes a query automaton generation unit for generating a query automaton from a path expression in which a retrieval condition for the database is described.

Murata, however, teaches **"wherein the access rights analysis device includes a query automaton generation unit for generating a query automaton**

from a path expression in which a retrieval condition for the database is described" as "A selection query is select (e1, e2) where e1 is a hedge regular expression and e2 is a pointed hedge representation" (Pages 132-133, Section 6.1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Murata's** would have allowed **Damiani's** and **Deo's** to provide a method to allow for further processing of output relations by enabling schema translations via queries for xml documents, as noted by **Murata** (Pages 126-127, Section 1).

Regarding claim 12, **Damiani** further teaches a database retrieval system comprising:

A) a path expression extraction unit for extracting the path expressions from a query expression specifying a retrieval method for the database (Pages 185-186); and

B) a query expression access right decision unit for deciding access rights in the database retrieval by the query expression based on decision results of access rights, which are obtained by the access right decision unit, for the individual path expressions extracted from the query expression (Pages 190-191).

The examiner notes that **Damiani** teaches **"a path expression extraction unit for extracting the path expressions from a query expression specifying a retrieval method for the database"** as "object is either a URI in Obj or is of the form URI:PE, where URI ε Obj and PE is a path expression on the tree of document URI" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"a query expression access right decision unit for deciding access rights in the database retrieval by the query expression based on decision results of access rights, which are obtained by the access right decision unit, for the individual path expressions extracted from the query expression"** as "The value of n.veclabel[t].sign can be "+" for permission, "-" for denials, and "ε" for no authorization" (Page 188, Section 6.1: Document Tree Labeling) and "Signs + and – must then be mapped to the other two values, namely 1 (true) and ½ (indeterminate" (Page 190, Section 6.1: Document Tree Labeling) and "As a result of the labeling process, the value of finlabel for each node n contains the sign, if any, reflecting whether the node can be accessed (+) or not (-)" (Page 191, Section 6.2: Transformation Process).

Regarding claim 15, **Damiani** teaches an access rights analysis method comprising:

B) generating an access control automaton from an access control policy in which an access control rule is described (Pages 185-186); and

C) storing the access control automaton in a predetermined storage means (Page 185)

D) performing logic operations related to the query automaton and the access control automaton, which are stored in the predetermined storage means (Page 191);

E) deciding an access right in database retrieval using the path expression (Pages 183 and 186, Figure 5)

The examiner notes that **Damiani** teaches **"generating an access control automaton from an access control policy in which an access control rule is described"** as "An access authorization a ε Auth is a five-tuple of the form: <subject, object, action, sign, type>" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that it is common knowledge that an automaton is a five-tuple with states, symbols, and transition states. The examiner further notes that Damiani teaches "storing the access control automaton in a predetermined storage means" as "At each server, a set of Auth of access authorizations specifies the actions that subjects are allowed (or forbidden) to exercise on the objects stored at the server" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"performing logic operations related to the query automaton and the access control automaton, which are stored in the predetermined storage means"** as "In particular, the final sign finlabel of each node n is determined as the result of operation ⊕ between the sign field of components of array n.veclabel considered in their priority order: LDH (local hard), RDH (recursive hard), L (local), R (recursive), LD (local, schema level), RD (recursive, schema level), LS (local soft), and

RS (recursive soft)" (Page 191, Section 6.1: Document Tree Labeling). The examiner further notes that **Damiani** teaches **"deciding an access right in database retrieval using the path expression"** as "Authorizations specified on an element can be defined as applicable to the element's attributes only (local authorizations) or, in a recursive approach, to its subelements and their attributes (recursive authorizations)" (Page 183, Section 5.1: Basic Features of the Access Authorizations) and "Figure 5 lists the resulting authorizations" (Page 186, Section 5.2: Access Authorizations).

Damiani does not explicitly teach:

E) without checking the XML document stored in the database.

Deo, however, teaches **"without checking the XML document stored in the database"** as "One or more applications may access individual volatile files 122 using the APIs 202. For instance, suppose an application would like to open a volatile file named "File1". In an implementation using Windows-brand operating system, the application calls a function named "SewCreateFile( )", which opens an existing file or creates a new file if one does not exist. In response to this function call, the file system 118 initially visits the ACL table 204 to determine whether the application has sufficient authorization to access the requested file. This is represented by flow arrow 230 in FIG. 2. Assuming the application is authorized, the file system 118 proceeds to the memory region directory 206 to determine which memory region holds the requested file (flow arrow 232). In this example, the directory 206 indicates that the file "File1" is located in RAM 106. The memory region directory 206 directs reference to the RAM-based file allocation table 212 (flow arrow 234), which provides physical location information to locate the volatile file 122(1) within RAM 106 (flow arrow 236)" (Column 5, lines 25-44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Deo's** would have allowed **Damiani's** to provide a method to protect files against from rouge or malicious agents, as noted by **Deo** (Column 1, lines 51-54).

Damiani and **Deo** do not explicitly teach:

A) generating a query automaton from a path expression in which a retrieval condition for the database is described.

C) storing the generated query automaton in a predetermined storage means.

Murata, however, teaches **"generating a query automaton from a path expression in which a retrieval condition for the database is described"** as "A selection query is select (e1, e2) where e1 is a hedge regular expression and e2 is a pointed hedge representation" (Pages 132-133, Section 6.1) and **"storing the generated query automaton in a predetermined storage means"** as "we construct match-identifying hedge automata form hedge regular expressions and pointed hedge representations" (Page 127, Section1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Murata's** would have allowed **Damiani's** and **Deo's** to provide a method to allow for further processing of output relations by enabling schema translations via queries for xml documents, as noted by **Murata** (Pages 126-127, Section 1).

Regarding claim 17, **Damiani** teaches a program comprising:

B) an access control automaton generation means for generating an access control automaton from an access control policy in which an access control rule is described (Pages 185-186); and

C) a logic operation means for deciding access rights in database retrieval using the path expression by performing logic operations related to the generated query automaton and access control automaton (Page 191).

The examiner notes that **Damiani** teaches **"an access control automaton generation means for generating an access control automaton from an access control policy in which an access control rule is described"** as "An access authorization a ε Auth is a five-tuple of the form: <subject, object, action, sign, type>" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that it is common knowledge that an automaton is a five-tuple with states, symbols, and transition states. The examiner further notes that **Damiani** teaches **"a logic operation means for deciding access rights in database retrieval using the path expression by performing logic operations related to the generated query automaton and**

access control automaton" as "In particular, the final sign finlabel of each node n is determined as the result of operation ⊕ between the sign field of components of array n.veclabel considered in their priority order: LDH (local hard), RDH (recursive hard), L (local), R (recursive), LD (local, schema level), RD (recursive, schema level), LS (local soft), and RS (recursive soft)" (Page 191, Section 6.1: Document Tree Labeling).

Damiani does not explicitly teach:

D) without accessing said data file.

Deo, however, teaches **"without accessing said data file"** as "One or more applications may access individual volatile files 122 using the APIs 202. For instance, suppose an application would like to open a volatile file named "File1". In an implementation using Windows-brand operating system, the application calls a function named "SewCreateFile( )", which opens an existing file or creates a new file if one does not exist. In response to this function call, the file system 118 initially visits the ACL table 204 to determine whether the application has sufficient authorization to access the requested file. This is represented by flow arrow 230 in FIG. 2. Assuming the application is authorized, the file system 118 proceeds to the memory region directory 206 to determine which memory region holds the requested file (flow arrow 232). In this example, the directory 206 indicates that the file "File1" is located in RAM 106. The memory region directory 206 directs reference to the RAM-based file allocation table 212 (flow arrow 234), which provides physical location information to locate the volatile file 122(1) within RAM 106 (flow arrow 236)" (Column 5, lines 25-44).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Deo's** would have allowed **Damiani's** to provide a method to protect files against from rouge or malicious agents, as noted by **Deo** (Column 1, lines 51-54).

Damiani and **Deo** do not explicitly teach:

A) a query automaton generation means for generating a query automaton from a path expression in which a retrieval condition for the database is described.

Murata, however, teaches **"a query automaton generation means for generating a query automaton from a path expression in which a retrieval**

condition for the database is described" as "A selection query is select (e1, e2) where e1 is a hedge regular expression and e2 is a pointed hedge representation" (Pages 132-133, Section 6.1).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Murata's** would have allowed **Damiani's** and **Deo's** to provide a method to allow for further processing of output relations by enabling schema translations via queries for xml documents, as noted by **Murata** (Pages 126-127, Section 1).

Regarding claim 18, **Damiani** further teaches a program comprising:

A) causing the computer to function as a path expression extraction means for extracting the path expressions from a query expression specifying a retrieval method for the database (Pages 185-186); and

B) a query expression access right decision means for deciding access rights in the database retrieval by the query expression based on decision results of access rights for the individual path expressions extracted from the query expression (Pages 190-191).

The examiner notes that **Damiani** teaches **"causing the computer to function as a path expression extraction means for extracting the path expressions from a query expression specifying a retrieval method for the database"** as "object is either a URI in Obj or is of the form URI:PE, where URI ε Obj and PE is a path expression on the tree of document URI" (Page 185, Section 5.2: Access Authorizations). The examiner further notes that **Damiani** teaches **"a query expression access right decision means for deciding access rights in the database retrieval by the query expression based on decision results of access rights for the individual path expressions extracted from the query expression"** as "The value of n.veclabel[t].sign can be "+" for permission, "-" for denials, and "ε" for no authorization" (Page 188, Section 6.1: Document Tree Labeling) and "Signs + and – must then be mapped to the other two values, namely 1 (true) and ½ (indeterminate"

(Page 190, Section 6.1: Document Tree Labeling) and "As a result of the labeling process, the value of finlabel for each node n contains the sign, if any, reflecting whether the node can be accessed (+) or not (-)" (Page 191, Section 6.2: Transformation Process).

14. Claims 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Damiani et al.** (Article entitled "A Fine-Grained Access Control System for XML Documents", dated May 2002) in view of **Deo et al.** (U.S. Patent 6,970,891) as applied to claims 6, 8-10, 13-14, 16, and 19-20, and further in view of **Hunnicutt et al.** (U.S. PGPUB 2003/0191846).

15. Regarding claim 21, **Damiani** and **Deo** do not explicitly teach a database retrieval system comprising:

A) wherein, if said access rights decided by said preliminary access rights analysis device is indeterminate, said database retrieval system retrieving said XML document to determine access rights.

Hunnicutt, however, teaches **"wherein, if said access rights decided by said preliminary access rights analysis device is indeterminate, said database retrieval system retrieving said XML document to determine access rights"** as "A further example of the general operation of the methods of the present invention is described with respect to User 2 requesting to read file-object 300. User 2, as one of clients 1-N, attempts to log onto server 100 by supplying a user-name and a password to server 100 over network 102. The user-name and password supplied by User 2 are recognized by Server 100 and User 2 is therefore allowed to log on to Server 100. Server 100 first checks user-token cache 200 of FIG. 2 for a user-token matching User 2's user-name. Entry 202 of user-token cache 200 matches User 2's user-name and therefore server 100 retrieves Token2 from user-token cache 200 as the user-token for User 2. User 2, now logged on to server 100, places a request to read file-object 300. Server 100 checks access-cache 400 for an access-permission matching the current request. Access-permissions 403 and 405 each have file-object 300 in the file-name field 401 but neither access-permission 403 or 405 have a user-token field 402 that matches User 2's user-token. This means that User 2 has not previously read file 300 and the system must perform a full, file open, access check. Referring to FIG. 3, file-object 300 is opened and access control list 301 is read to determine the access-permission granted to User 2. ACE 306 defines User 2's granted access to file-object 300. Permitted-access field 304 of ACE 306 indicates that User 2 has read/write permission for file 300. Access for reading file-object 300 is therefore provided to User 2. Once the file-open access check is completed, an appropriate access-permission (not shown) is added to access-cache 400 so that a file-open access check will not need to be performed the next time User 2 requests file-object 300" (Paragraphs 40-41).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Hunnicutt's** would have allowed **Damiani's** and **Deo's** to provide a method for an access check system that performs the necessary access check, even at the file-level of access control, without the relatively slow operation of opening the requested file-object to check the associated access control list, as noted by **Hunnicutt** (Paragraph 8).

Response to Arguments

16. Applicant's arguments with respect to claims 1-21 have been considered but are moot in view of the new ground(s) of rejection (**Deo** with respect to the preliminary access rights decision without accessing a document) and (**Hunnicutt** with respect to downloading a document if an indeterminate condition is reached).

17. Applicant's arguments filed 12/06/2007 have been fully considered but they are not persuasive.

Applicants argue on page 3 that **"Damiani teaches away from the first type of access control system discussed above, which operates at the file-system level, independent o the data that is being protected"**. However, Applicants are also reminded that in order to disqualify a reference based on a "teach away" reasoning, the reference has to *explicitly* suggest or disclose the so-called teach away steps - Applicants assertion can not be accepted if it is unsupported by a valid evidence. In this case, the added amendments that state that access control is performed before the accessing of the document is taught by the new secondary reference of **Deo**. The primary reference of **Damiani** teaches the access control steps and states that an independent access control system can also be realized in its background. Therefore, **Damiani** does not teach away from the instant invention.

Applicants argue on page 5 that **"Claim 10 also states that the access rights decision "is one of…indeterminate". There is no teaching of Damiani of these three possible decisions...In fact, Damiani does not teach that an indeterminate decision is possible nor how such an occurrence would be handled"**. However, the examiner wishes to refer to pages 188 and 190 of **Damiani** which state "The value of n.veclabel[t].sign can be "+" for permission, "-" for denials, and "ε" for no authorization" (Page 188, Section 6.1: Document Tree Labeling) and "Signs + and – must then be mapped to the other two values, namely 1 (true) and ½ (indeterminate" (Page 190, Section 6.1: Document Tree Labeling). The examiner further wishes to state that a positive sign teaches "always permitted", the negative sign teaches "always denied", and the "ε" sign broadly teaches indeterminate.

Conclusion

18. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Article entitled "Regulating Access to XML documents" by **Gabillon et al.** on July 2001. The subject matter disclosed therein is pertinent to that of claims 1-21 (e.g., methods to control access to XML documents)

Article entitled "Efficient Filtering of XML Documents for Selective Dissemination of Information" by **Altinel et al.** in 2000. The subject matter disclosed therein is pertinent to that of claims 1-21 (e.g., methods to control access to XML documents)

U.S. PGPUB 2004/0172234 issued to **Dapp et al.** on 02 September 2002. The subject matter disclosed therein is pertinent to that of claims 1-21 (e.g., methods to control access to XML documents)

U.S. PGPUB 2003/0229852 issued to **Uramoto et al.** on 12 December 2003. The subject matter disclosed therein is pertinent to that of claims 1-21 (e.g., methods to control access to XML documents)

U.S. PGPUB 2004/0073870 issued to **Fuh et al.** on 25 March 2004. The subject matter disclosed therein is pertinent to that of claims 1-21 (e.g., methods to control access to XML documents)

Contact Information

19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner
Art Unit 2168

/Mahesh H Dwivedi/
Examiner, Art Unit 2168
03/11/2008

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168